Recognizing and Tagging Temporal Expressions in Spanish

Estela Saquete, Patricio Martínez-Barco and Rafael Muñoz
Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante
Estela.Saquete@ua.es
{patricio, rafael}@dlsi.ua.es

Abstract

This paper shows a system about the recognition of temporal expressions in Spanish and the resolution of their temporal reference. For the identification and recognition of temporal expressions we have based on a temporal expression grammar and for the resolution on an inference engine, where we have the information necessary to do the date operation based on the recognized expressions. For further information treatment, the output is proposed by means of XML tags in order to add standard information of the resolution obtained. Different kinds of annotation of temporal expressions are explained in another articles [WILSON2001][KATZ2001]. In the evaluation of our proposal we have obtained successful results.

1. Introduction

The study of anaphora phenomena has been carried out for a lot of researches. Most of these researches have focused on pronominal anaphora and a few of them on definite descriptions. Most of temporal expressions could be considered as a type of definite description, but a few of them are temporal adverbs like “mañana” (*tomorrow*). The research work developed in definite description is focused on establishing a relationship between anaphoric expressions and their antecedents. In these work, if the definite description is a temporal expression it has been only solved establishing the relationship but not inferring the new date. The resolution of temporal expressions involves the recognition of them and the inference of the new date.

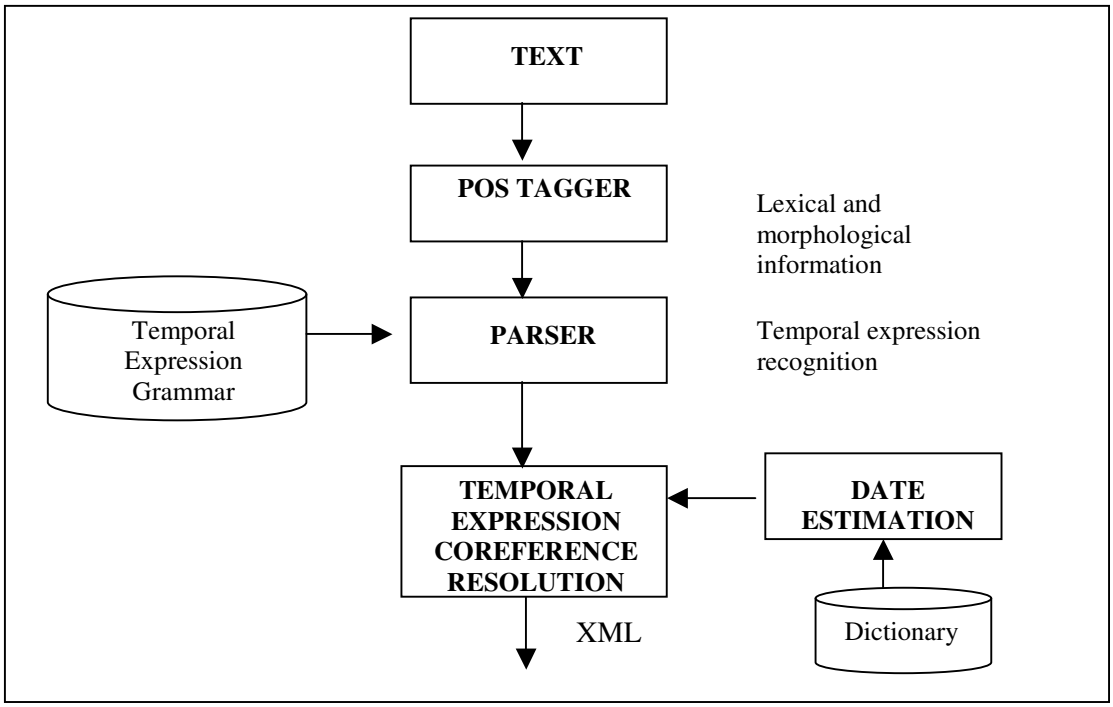

Fig. 1. Graphic representation of the system proposed

Work focused on temporal expression should to solve both tasks. In the literature we can found several studies focused on temporal expressions (Guillen et *al.* 1995) (Wiebe et *al.* 1998). These studies are based on the use of a temporal model that is able to interpret different formats for date and time expressions. Some of them are based on the application of empirical methods using the focus theory proposed in (Grosz et al. 1986).

In this paper a proposal of a grammar for the recognition of temporal expressions in Spanish is presented, as well as an approximation to the resolution of the coreference introduced by them, as is explained in (Saquete and Martínez-Barco 2000). Moreover, in this paper a set of tags is used to annotate the temporal expressions in the corpus.

In a text, there are dates with typical representations like, for example: “23/01/2000” o “23 de enero del 2000” (*23rd of January of 2000*), but we can find references to dates named previously too, for example: “dos días antes” (*two days before*), “la semana anterior” (*the previous week*), etc. This kind of coreference should be solved and mapped to dates with a standard format for a more efficient analysis of the text. For that, we use a grammar for the recognition of temporal expressions with their correspondent temporal parser, and an inference engine to solve and to map these expressions in a standard format: mm/dd/yyyy (hh:mm, for time expressions). Once the kind of reference and the interpretation of the expression are solved, the text is tagged with XML tags.

2. System structure

The system proposed is shown in Figure 1. The system has the plain text as input. These texts are tagged with lexical and morphological information using the POS-tagger developed by Pla (Pla 2000) and this information is the input of the temporal parser. The temporal parser is implemented using an ascending technique and it is based in a temporal grammar shown below. Once the temporal expressions are recognized, these are introduced into the resolution unit called Temporal Expression Coreference Resolution, which will update the value of the reference according to the date that it is referring to, and then it will generate the XML tags for each expression.

There are two different kinds of rules that are used for the grammar because there are two different kinds of temporal expressions too:

1. There are anaphoric and not anaphoric expressions. That is why there are rules for the date and time recognition (non-anaphoric expressions like “12/06/1975”).

2. There are rules for the temporal reference recognition (anaphoric temporal expressions that need another complete temporal expression to be understood “*two days before*”).

Temporal references could be divided in two groups: time adverbs (i.e. *yesterday, today*) and nominal phrases that refer to temporal relationships (i.e. *two years before*).

Tables 1 and 2 show some rules used for the recognition of dates and the detection of anaphoric temporal expressions, respectively.

3. Coreference resolution based on a temporal model

Previous to coreference resolution, the parser identifies temporal expressions. Temporal expressions can be anaphoric or non-anaphoric. For this reason, we have split the rules for the identification or recognition of temporal expression in two different sets. The first set is made up by the rules for the identification of non-anaphoric temporal expressions (table 1) and the second one is made up by the rule to recognize the anaphoric temporal expressions (table 2). The coreference module should be applied to the temporal expressions recognized by the rules of the second set.

For the coreference resolution we use an inference engine that contains the interpretation for every reference named before. If we compare this system with traditional anaphoric systems, the algorithm for the treatment of temporal expressions needs to carry out an additional step. In our algorithm to solve the coreference of anaphoric temporal expressions, two different tasks can be distinguished:

1. Looking for the antecedent. This task is similar to the traditional approach to anaphora resolution. The algorithm chooses the antecedent from a list of candidates. Two main candidates are usually chosen. The first candidate is related to the date of the text, i.e. the date when the newspaper was written. This date is considered as *default date,* called in this paper *FechaP*. The second candidate is the previous non-anaphoric temporal expressions, called in this paper *FechaAnt* (*two days before* has as antecedent the previous date in the text). If none is found the default date is considered as antecedent. Sometimes, the temporal expression includes prepositional phrase with information about and event or process (*the day of the final match*), in this case the algorithm should look for the date associated to the event or process from the list of candidates. The following process is carried out:

    3. By default, the newspaper’s date is used as a base referent (temporal expression) if it exits. If not, the system date is used. (“ayer” *(yesterday)* Day(FechaP) –1 / Month(FechaP) / Year(FechaP).

    4. In case of finding a non-anaphoric temporal expression, it is stored as FechaAnt storing the old FechaAnt in a list of candidates. This value is updated every time that a non-anaphoric temporal expression appears in the text.

2. Providing the new date. Once the antecedent is selected, the new date should be inferred. This new step is related to provide a new date or time. The references are estimated using the antecedent selected in the previous step. This model is based on the two rules below and it is only applicable to these dates that are not *FechaP*, since for *FechaP* there is nothing to resolve

In Table 3 some of the entries of the dictionary used in the inference engine are shown. Moreover, the inference engine has the correspondence between numeric and string expressions of days and months, that is, *one* have the value *1* and *July* is *07*.

The module that makes the estimation of the dates will accede to the right entry in the inference engine in each case and it will apply the function specified obtaining a date in the format *mm/dd/yyyy or* a range of dates. So, at that point the coreference will have been resolved.

date dd + ″ /″ + mm + “/” + (yy)yy (12/06/1975) *(06/12/1975)*
date dd + “-“ + month + “ –“+ (yy)yy (12-junio-1975) *(12th-June-1975)*
date dd + “de” + mm + “de” + (yy)yy (12 de junio de 1975) *(12th of June of 1975)*
date (“El”) + day_of_week+ dd + ″de″+ month + “de” + (yy)yy (El domingo 12 de junio de 1975) *(Sunday, 12th of June of 1975)*
date month+ ″de″+ yy(yy) (Febrero de 1975) *(February of 1975)*
date dd + ″de″+ month + “de” + (yy)yy + “a las”+ time (12 de junio de 1975 a las 6 y media) *(12th of June of 1975 at half past six)*
dd [“01”|”02”|”03”|….|”31”]
day [“uno”| ″dos″ |… |”treinta y uno”] *[one|two|…|thirty one]*
mm [“01”|”02”|”03”|…|”12”]
month [“enero”| “febrero”| “marzo”	“abril”	“mayo”	“junio”	“julio”	“agosto”	“septiembre”	“octubre”	“noviembre”	“diciembre”] *(January|February|March|April|May|June|July|August|September|October|November|December)*
a [“1”	“2”	“3”	…”9”	“0”]
day_of_week [“lunes”	“martes”	“miércoles”	“jueves”	“viernes”	“sábado”	“domingo”] *(Monday|Tuesday|Wednesday|Thursday|Friday|Saturday|Sunday)*
time [hh:mm(:ss)	hh (y	menos) cuarto	hh y media	...]

Table 1. Sample of rules for the date recognition

Time Adverbs	reference “ayer” *(yesterday)* reference “mañana” *(tomorrow)* reference “anteayer” *(the day before yesterday)* reference “anoche” *(last night)*
Temporal Nominal Phrases	reference ”el”+ “próximo” + [“día”	“mes”	“año”] *(the next day|month|year)* reference “un” + [“día”	“mes”	“año”] + “después” *(a day|month|year later)* reference num + [“días”	“meses”	“años”] + “después” *(num days|months|years later)* reference “un” + [“día”	“mes”	“año”] + “antes” *(a day|month|year before)* reference num + [“días”	“meses”	“años”] + “antes” *(num days|months|years before)* reference “dentro” + “de” + “un” + [“día”	“mes”	año”] *(within a day|month|year)* reference “dentro” + “de” + num +[“días”|“meses”| “años”] *(within num days|months|years)* reference “el” + “pasado” + [“día”	“mes”	“año”] *(the last day|month|year)* reference “el” + [“día”	“mes”	“año”] + “siguiente” *(the next day|month|year)* reference “los” + num + [“días”	“meses”	“años”] + “siguientes” *(the num next days|months|years)* reference “el” + [“día”	“mes”	“año”] + “pasado” *(the last day|month|year)* reference “los” + num + [“días”	“meses”	“años”] +“pasados” *(the last num days|months|years)* num [“dos”	“tres”	“cuatro”	“cinco”	….] *(two|three|four|five|…)*

Table 2. Sample of rules for the reference recognition

REFERENCE	DICCIONARY ENTRY
"ayer" *(yesterday)*	Day(FechaP) –1 / Month(FechaP) / Year(FechaP)
"mañana" *(tomorrow)*	Day(FechaP) +1 / Month(FechaP) / Year(FechaP)
"anteayer" *(the day before yesterdary)*	Day(FechaP) –2 / Month(FechaP) / Year(FechaP)
"anoche" *(last night)*	Day(FechaP) –1 / Month(FechaP) / Year(FechaP) [09:00-05:00]
"el"+ "próximo"+"día" *(the next day)*	Day(FechaP)+1 / Month(FechaP) / Year(FechaP)
"un"+"mes"+"después" *(a month later)*	[DayI/Month(fechaAnterior)+1/Year(fechaAnterior)-- DayF/Month(fechaAnterior) +1/ Year(fechaAnterior)]
num+"años"+ "después" *(num years later)*	[01/01/ Year(fechaAnterior) + num -- 31/12/ Year(fechaAnterior) + num]
"un" + "día" + "antes" *(a day before)*	Day(fechaAnterior)- 1/Month(fechaAnterior)/Year(fechaAnterior)
num+"meses"+ "antes" *(num months before)*	[DayI/Month(fechaAnterior) -num / Year(fechaAnterior) – DayF/ Month(fechaAnterior) - num / Year(fechaAnterior)]
"dentro"+"de"+"un"+ "año" *(within a year)*	[01/01/ Year(fechaAnterior) +1 – 31/12/ Year(fechaAnterior) +1]
"dentro"+"de"+num+ "días" *(within num days)*	Day(fechaAnterior)+num / Month(fechaAnterior) / Year(fechaAnterior)
"el" + "pasado" + "día" *(the last day)*	Day(fechaAnterior)-1/Month(fechaAnterior) / Year(fechaAnterior)
"el"+"mes"+"siguiente" *(the next month)*	[DayI / Month(fechaAnterior) +1 / Year(fechaAnterior) -- DayF / Month(fechaAnterior) +1 / Year(fechaAnterior)]
"los"+num+"años"+ "siguientes" *(the num years later)*	[01/01/Year(fechaAnterior) -- 31/12 / Year(fechaAnterior) +num]
"el" + "día" + "pasado" *(the last day)*	Day(fechaAnterior)-1/Month(fechaAnterior) / Year(fechaAnterior)
"los"+num+"meses"+ "pasados" *(the num last months)*	[DayI/Month(fechaAnterior) - num / Year(fechaAnterior) – DayF/Month(fechaAnterior) – 1 / Year(fechaAnterior)]

Table 3. Sample of some of the entries of the dictionary

4. Tagging of temporal expressions

Several proposals for the annotation of temporal expressions have been arisen in the last few years (Wilson et *al.* 2001) (Katz and Arosio 2001) since this kind of research has started. In this section, we proposed doing this annotation using XML tags, in order to standardize anaphoric and non-anaphoric temporal expressions.

4.1. XML

In our proposal we have chosen XML to define the set of tags we are going to use. XML stands for *eXtensible Markup Language* and it provides a subset of the SGML (*Standard Generalized Markup Language*). XML offers a non-ambiguous text-based method to develop data structures. XML documents represent data by means of tags.

XML was developed by a Generic SGML Editorial Review Board formed under the auspices of the W3 Consortium in 1996 and chaired by Jon Bosak of Sun Microsystems, with the participation of a Generic SGML Working Group also organized by the W3C (W3C 2002). Since then, the use of XML has been generalizing until becoming the universal standard for data electronic exchange.

XML has specific rules that must be strictly followed in order to make a new document. The XML document must fulfill the set of constraints being established in the *Document Type Declaration* (DTD). This DTD contains the structure of the document, and the validity of a XML document could be tested through this DTD: a well-formed document is valid only if it contains a proper document type declaration and if the document obeys the constraints of that declaration (element sequence and nesting is valid, required attributes are provided, attribute values are of the correct type, etc.).

As a result, XML provides us several advantages to our proposal:

- Both persons and machines easily interpret it. As a consequence that makes easy both the manual and the automatic extraction of dates from a text.
- XML documents are easily tagged from an automatic process, but also a manual annotator could make use of commercial XML editors to develop this task.
- XML is standard.
- A DTD has been built to check the validity of each XML document. So manual and automatic annotations can be automatically tested looking for possible mistakes.

4.2. Annotation schema

An appropriate annotation schema has been defined to mark every temporal expression found. This schema is base on the following ideas:

First, every temporal expression is going to be marked. That includes the markup of dates and times that could be expressed in whatever format, including anaphoric and not anaphoric expression. In this way, the following rules are going to be applied:

a) Full date expressions, that is, non-anaphoric temporal expressions are going to be marked using the standard format for dates: mm/dd/yyyy.

10 de abril de 2002 *04/01/2002*
(*10th of April of 2002*)

b) Full date and time expressions (again, non-anaphoric expressions) are going to be marked in the same way, including in this case the time parameter, in the standard 24-hour format: hh:mm

10 de abril de 2002, a las nueve
04/10/2002, 09:00
(*10th of April of 2002, at nine o'clock*)

c) Time expressions (without explicit date, but referring to an omitted date) are anaphoric expressions. Then, the coreference resolution module is applied before tagging. Once the date of the time is calculated, date and time are tagged.

A las nueve *04/10/2002, 09:00*
(*At nine o'clock*)

d) Some kind of time expressions without explicit date are not anaphoric expressions because they do not refer to an omitted date. In this case only the time parameter is need, so the coreference resolution module is not used.

Todos los días a las nueve *09:00*
(*Everyday at nine o'clock*)

e) Anaphoric date expressions need the coreference resolution module to define the absolute expressions to which they refer to. After that, the appropriate tag will be marked.

El próximo miércoles *04/17/2002*
(*Next Wednesday*)

f) Anaphoric date and time expressions follow the previous rule, calling the coreference resolution previous to be marked.

El próximo miércoles, a las nueve
04/17/2002, 9:00
(*Next Wednesday, at nine o'clock*)

g) Non-anaphoric ranges of dates and/or time are directly tagged by means of the initial and the final date.

Del 10 de abril al 20 de abril de 2002
04/10/2002 — 04/20/2002
(*From 10 of April to 20 of April of 2002*)[1]

El 10 de abril, de 9 a 11 y media
04/10/2002, 9:00 — 11:30
(*10th of April from 9 to half past 11*)

Del 10 de abril de 2002 a las nueve al 20 de abril de 2002 a las doce
04/10/2002, 9:00 — 04/20/2002, 12:00
(*From 10 of April of 2002 at nine o'clock to 20 of April of 2002 at twelve o'clock)*

h) Anaphoric ranges of dates and/or time are previously solved using the coreference resolution module. Then the full date is marked.

[1] Direct translation from Spanish

Del miércoles al jueves 04/17/2002 — 04/18/2002
(*From Wednesday to Thursday*)

i) What we have called *fuzzy temporal expressions* have not a concrete date or time related to. For this reason the coreference resolution module is useless. However, in order to be identified as temporal expressions we decided to mark them using a special parameter with the "FUZZY" value.

The grammar used to identify anaphoric and not anaphoric expressions acts as a trigger launching the appropriate rule in each case.

4.3. Tag definitions

The structure of tags used to define temporal expression data is the following:

```
< DATE_TIME TYPE="value"
        VALDATE1="value"
        VALTIME1="value"
        VALDATE2="value"
        VALTIME2="value" >
        Expression
</DATE_TIME>
```

In this structure the next elements are used:

- DATE_TIME is the name of the tag for non-anaphoric temporal expressions.
- VALDATE# store the range of dates obtained from the inference engine.
- VALTIME# store the range of times obtained from the inference engine.
- TYPE attribute could have the following values: CONCRETE, PERIOD and FUZZY:
  - § CONCRETE is referring to only a date.
  - § PERIOD is referring to a period of time.
  - § FUZZY attribute is used when we really do not know the date or period of time when a temporal expression is referring to.

Moreover, VALDATE1, VALDATE2, VALTIME1 and VALTIME2 are optional attributes:

- VALDATE2 and VALTIME2 are used to establish ranges. So, if we try to tag a concrete date (TYPE adopt the value CONCRETE) then these attributes are omitted.
- VALTIME1 could be omitted if only a date is specified.
- VALDATE1 could be omitted if only a time must be specified. This is the case in which the date does not mind. For example, *todos los días a las nueve* (*everyday at nine o'clock*). However, when only a time expression is specified, such as *a las nueve* (*at nine o'clock*), the VALDATE1 of this time must be computed.

The use of XML allows us to take advantage of the XML schema in which the tag language is defined. This schema let an application know if the XML file is valid and well-formed. The schema defines the different kind of elements, attributes and entities that are allowed, and can express some limitations to combine them. Moreover, use the same syntax as XML and the schemas are extensible. Once the XML file has been generated, a parser of our XML needs to be defined to make the information useful.

Tables 4a and 4b show several examples for tagging temporal expressions (non-anaphoric and anaphoric). Table 5 shows an example of an annotated text in which the features of the used tags are shown. In this example we assume that the newspaper's date is 04/25/2000. The system, for the reference "el próximo año"*(the next year),* will return "01/01/2001-12/31/2001". For the reference "mañana" *(tomorrow)* it will return 04/26/2000.

ABSOLUTE TAGS (non-anaphoric temporal expressions)
<DATE_TIME VALDATE1="06/12/2001">12 de junio de 2001</DATE_TIME>
<DATE_TIME VALDATE1="06/12/2001" VALTIME1="20:30">12 de junio de 2001 a las ocho y media de la tarde</DATE_TIME>
<DATE_TIME VALDATE1="06/12/2001" VALTIME1="20:00" VALDATE2="06/12/2001" VALTIME2="21:00" >12 de junio de 2001 entre las ocho y las nueve de la tarde</DATE_TIME>

Table 4a. Sample of the tags generated by this system

REFERENCE TAGS (anaphoric temporal expressions)
<DATE_TIME_REF VALDATE1="06/11/2002">ayer</DATE_TIME_REF> <DATE_TIME_REF VALDATE1="01/01/2002" VALDATE2="12/31/2007">los 5 años siguientes</DATE_TIME_REF>

Table 4b. Sample of the tags generated by this system

"La oficina de Congresos de la Universidad ha propuesto 5 congresos para **<DATE_TIME_REF TYPE=** "PERIOD" **VALDATE1="01/01/2000" VALDATE2= "12/31/2000>** este año**</DATE_TIME_REF>,** sin embargo, el crecimiento para **<DATE_TIME_REF TYPE=**"PERIOD" **VALDATE1= "01/01/2001" VALDATE2="12/31/2001>** el próximo año**</DATE_TIME_REF>** será superior a los 15. Por otro lado, el Director de la oficina ofrece **<DATE_TIME_REF TYPE=** "CONCRETE" **VALDATE1= "04/26/2000" >** mañana**</DATE_TIME_REF>** una conferencia."

(The University Conference Office has proposed 5 c onferences for **<DATE_TIME_REF TYPE=**"PERIOD" **VALDATE1="01/01/2000" VALDATE2="12/31/2000>** *this year* **</DATE_TIME_REF>**, *however, the increase for* **<DATE_TIME_REF TYPE=**"PERIOD" **VALDATE1= "01/01/2001" VALDATE2="12/31/2001>** *the next year* **</DATE_TIME_REF>** *will be over 15. On the other hand, the Office Man ager offers* **<DATE_TIME_REF TYPE=** "CONCRETE" **VALDATE1= " 04/26/2000" >** *tomorrow* **</DATE_TIME_REF>** *a lecture).*

Table 5. Example of annotated text

5. System evaluation

For implementing the system we need two different units, it is necessary to implement a parser and for that we have used LPA Prolog, because this language is based on rules as the parser is. The other unit is implemented in Visual Basic because this language has several time functions. This unit generated the XML tags too. The implementation of a XML parser is an optional possibility. The evaluation of the system has been done with a sample extracted from 16 articles that belong to the digital edition on the Internet of two Spanish newspapers describing different topics. The results obtained for the articles showed a precision and a recall of 95.59 % and 82.28% respectively.

The total has been calculated according to the number of successes being 195, the number of treated references is 204 and the number of total references is 237.

5.1. Error analysis

However, some fails in the system have been detected and we show their possible improvements below:

- The unit that resolves temporal references is not able to resolve undetermined temporal references like "hace unos cuantos días" (*some days before*) accurately. Here, one possible solution is the use of the semantic information. For example, if the sentence is "*some days before*", the system will suppose that is less time than a week, because we usually use the word "*week*" referring to seven days.

- It is possible that we have non-anaphoric expressions that make reference to an event or a fact and, despite they are not temporal expressions themselves, they mean a date or period of time too. For example: "ganó el mundial y al día siguiente se lesionó" (*he won the World Champion and the next day he hurt himself*).

- In newspaper articles, sometimes we find expressions like "el sábado hubo un accidente" (*Saturday there was an accident*). To resolve this expression we should know some extra information of the context where the reference is. This extra information could be the sentence verb. If it is a past verb that means that the sentence is

referring to *the last Saturday*. However, if the verb is future, it is referring to *the next Saturday*. In our system, we are not using this kind of information, so we assume that this kind of reference if referring to the last day, not the next, because the news usually tells us facts occurred previously.

6. Conclusions

In this paper a system for temporal expressions recognition in Spanish and their reference resolution has been presented, based on a temporal model proposed. The system has two different units: the parser based on a temporal expression grammar, which allows to identify these kind of expressions and a coreference resolution unit which is based in a inference engine and make a transformation of the expressions to dates, resolving their reference in this way. The evaluation of the system shows successful results of precision and recall for our proposal.

For future works, it is pretended to extend the system with the temporal references that are not treated in this paper. Moreover, the study of the verbal forms in the sentences where the references are found will improve the efficiency of the system solving some kind of expressions.

7. Acknowledgements

This paper has been supported by the Spanish Government (MCYT) under grant TIC2000-0664-C02-01/02.

8. References

Katz, G. and Arosio, F. (2001). The Annotation of Temporal Information in Natural Language Sentences. In *Proceedings of the Workshop On Temporal And Spatial Information Processing (ACL'2001)*.

Pla, F. (2000). Etiquetado Léxico y Análisis Sintáctico Superficial basado en Modelos Estadísticos. Ph D. Thesis. Departamento de Sistemas Informáticos y Computación. Universidad de Politécnica de Valencia.

Saquete, E. & Martinez-Barco, P (2000). Grammar specification for the recognition of temporal expressions. In *Proceedings of the*

Wilson, G., Sundheim, B., & Ferro, L. (2001). A Multilingual Approach to Annotating and Extracting Temporal Information. In *Proceedings of the Workshop On Temporal And Spatial Information Processing (ACL'2001)*.

W3C (2002). Extensive Markup Language web page. W3C World Wide Web Consortium. http:// www.w3.org/XML.